# CRAY X-MP及びCRAY-2における大規模数値計算ベクトル・パイプライン方式スーパーコンピュータの多重プロセッサ化の必要性とその応用(スーパーコンピュータのための数値計算アルゴリズムの研究)

AUTHOR(S):

加藤, 毅彦; 大吉, 哲郎; 三上, 和徳

数理解析研究所講究録
613巻 1987年 44-59


# CRAY X-MP及びCRAY-2における大規模数値計算
# ベクトル・パイプライン方式スーパーコンピュータ
# の多重プロセッサ化の必要性とその応用

日本クレイ(株) 加藤毅彦 ( Takehiko Kato )
大吉哲郎 ( Tetsuro Oyoshi )
三上和徳 ( Kazunori Mikami )

## 1. はじめに

科学技術計算においては、逐次演算、配列演算、入出力等の処理が混在し、それらすべてにわたって調和のとれた高速化により初めて本質的な処理時間の短縮化が可能となる。現在のスーパーコンピュータの主流であるベクトル・パイプライン方式は、配列演算の並列性に着目したものであり各種の離散化手法により帰着される行列演算の高速化に適している。また物理現象や科学技術計算に存在するより大規模な並列性に着目して多重プロセッサ方式とベクトル・パイプライン方式を併用したスーパーコンピュータも製品化が進められており、すでに現在世界で稼働中のスーパーコンピュータの約3割が多重プロセッサ構成となっている。

本稿では、スーパーコンピュータ CRAY X-MPとCRAY-2を例に上記の処理時間の短縮化に関連した多重プロセッサの必要性とその適用法について述べる。

## 2. スーパーコンピュータの性能

ベクトル・パイプライン方式のスーパーコンピュータの理論的最大性能は、ループ長無限大、ベクトル化率100パーセントの仮定のもと下記のように示される；

$$Pmax = M / C \qquad (1)$$

ここにPmax,C,Mは、それぞれ理論的最大性能，ベクトル演算器のパイプラインピッチもしくはクロック周期，及び並列実行可能な演算器の最大数である。理論的最大性能を向上させるためには、Cの最小化としてアーキテクチャ，回路設計，モジュール実装技術，冷却技術の改良による信号伝達時間の短縮化が必要であり、またMの最大化としては、多重演算器，多段パイプライン，多重プロセッサの採用が挙げられる。

またベクトル・パイプライン方式のスーパーコンピュータの実効性能は、有名なAmdahlの法則によって表されるがそれを積分し一般化することにより特定のベクトル化率領域における平均実効性能を下記のように表すことができる；

$$P = SR \ln (((a_2 + R(1 - a_2)) / ((a_1 + R(1 - a_1))) / ((1 - R)(a_2 - a_1)) \qquad a_1 \leq a \leq a_2 \qquad (2)$$

ここに P,S,R,aは、それぞれ実効性能，スカラー性能，最大加速率，ベクトル化率でありSRは、式(1)の理論的最大性能Pmax

に等しくなる。現実的には、多種多様なユーザーにより使用される場合において $0 \leqq a \leqq 1$ となるが、一般的な科学技術計算で $0.3 \leqq a \leqq 0.8$、高度に最適化されたアルゴリズムとプログラミング技法を使用して $0.5 \leqq a \leqq 0.95$、理想的な状態で $0.95 \leqq a \leqq 1.0$ を aの領域として選ぶことが適切であると考えられる。この $0.95 \leqq a \leqq 1.0$ を式(2)に適用すると理論的瞬間最大性能以上に本質的で意味のある実効最大性能(sustained performance)と呼ばれる持続性能の判定基準が得られる。

式(2)を現実的なベクトル化率領域のもとで検討すると、1)スカラー性能が実効性能の向上に支配的役割を果たす、2)多段パイプライン，多重演算器によるベクトル性能(最大性能)のみの向上すなわち最大加速率のみの向上は、実効性能への貢献が低いということが明らかとなる。例えば、実効性能を2倍もしくは4倍に向上させるためには、ベクトル性能のみを2倍もしくは4倍に高速化するのではなく、スカラー，ベクトル性能ともに 2倍もしくは4倍に高速化することが必要になってくる。そしてそのためには、1)スカラー，ベクトル演算器ともに 2倍，4倍に高速化する、2)スカラー，ベクトル演算器ともに 2重化，4重化するという方法が考えられる。スカラー，ベクトル演算器ともに実質的に高速化するためには、先に述べたようにアーキテクチャ，素子技術，回路技術，冷却技術等の改善によるスカラー，ベクトル演算器のクロック周期の短縮化が必要となる。現在これらの技術の進歩にはめざま

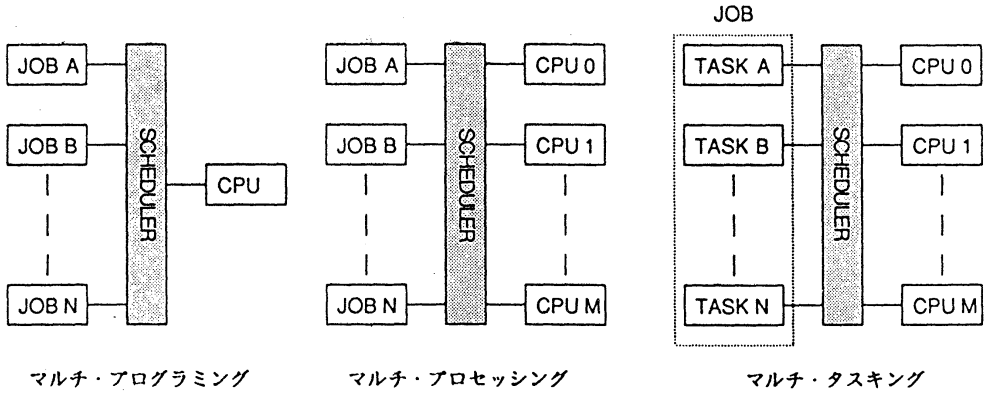
しいものがあり、その結果ベクトル、スカラーともに10ナノ秒以下のクロック周期を持つスーパーコンピュータも数多く稼働している。しかしながらより高速なコンピュータへの要望は、これらの技術の進歩以上に高く必然的にベクトル・パイプライン方式に加え、演算器の多重化をスカラー,ベクトル演算器ともに施すことが必要となってくる。演算器の多重化については、ひとつのプロセッサ内に加算器,乗算器等を複数ずつ配置する方法もあるがスーパーコンピュータの現実的な使用法がマルチ・ユーザー,マルチ・ジョブであり演算器資源の有効活用の見地からも多重プロセッサ方式の採用による高速化が最善の方法と考えられる。

多重プロセッサによる加速率Pmは、基本的にプロセッサ数とアルゴリズムの並列度に依存し、下記のWareの方程式によって表される;

$$Pm = T[\alpha(T/M) + (1-\alpha) + \theta(\alpha, M)] \qquad (3)$$

ここにT,$\alpha$,M,$\theta$は、それぞれ単プロセッサ時の実効時間,並列化率,プロセッサ数,オーバーヘッドである。もしオーバーヘッドが無視できれば式(3)は、Amdahlの法則と同型になる。多重プロセッサ方式の多段パイプライン，多重ベクトル演算器方式に対する利点は、複数ジョブの並列同時処理であるマルチ・プログラミング及びマルチ・プロセッシングと単一ジ

ョブの並列同時処理であるマルチ・タスキングの3つの並列多重処理を効率良くかつ柔軟に行える点にある(図－1)。




図－1　多重プロセッサによる並列処理

低並列度のジョブに対しては、疎結合システムとして自動ベクトル処理によるマルチ・プログラミング及びマルチ・プロセッシングを行い、高並列度のジョブには、自動ベクトル処理にマルチ・タスキングを併用することにより、どのようなユーザージョブに対しても常に演算器資源を充分に活用することが可能となる。マルチ・タスキングに関しては、効率的な活用のためにこれまでのアルゴリズムの再検討及び新しいアルゴリズムの研究が必要となるが 欧米ですでに多くの研究成果があげられている。

現実の大規模数値計算においては、スーパーコンピュータの演算器性能と同様に主記憶及び外部記憶装置の機能と性能が非常に大きな意味を持つ。スーパーコンピュータの主記憶装置は、汎用超大型機と較べても大容量で、高速化を優先するために仮想記憶方式ではなく実記憶方式を採用している。

しかしながら256メガ語(2ギガバイト)の主記憶容量を持つ後述のCRAY-2を除き現在のスーパーコンピュータといえども大規模なデータをすべて主記憶のみで処理することは容易でなく磁気ディスクのような外部記憶装置が必要となる。磁気ディスク装置単体の転送速度は毎秒3-12メガバイトの水準にあり演算処理装置の高速性に比較すると十分とは言えない場合もある。それを補うために、毎秒1ギガバイト以上の転送速度を持った半導体記憶装置(SSD)を外部入出力装置としてサポートしているスーパーコンピュータも多い。

これまでは、一般にCPU性能,特にベクトル性能のみに着目されがちであったが実際のスーパーコンピュータユーザーにとって最も切実な問題は、いかに短時間にジョブが処理されて手元に戻ってくるかという経過時間であり、そのためにはシステム全体の性能に着目しなければならない。図-2にスーパーコンピュータに用いられている上述の高速化技術による経過時

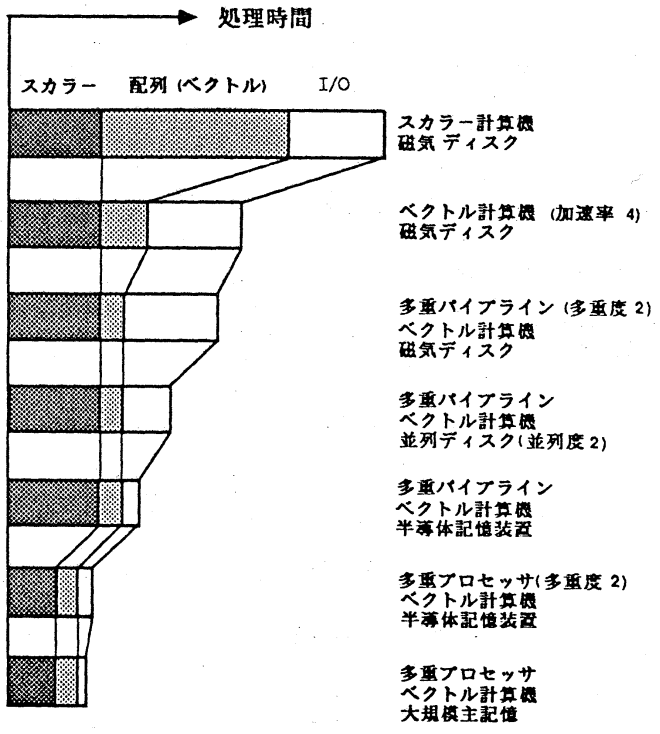



図-2 高速化技術による処理能力の向上

間の短縮化すなわち処理能力向上の様子を示す。

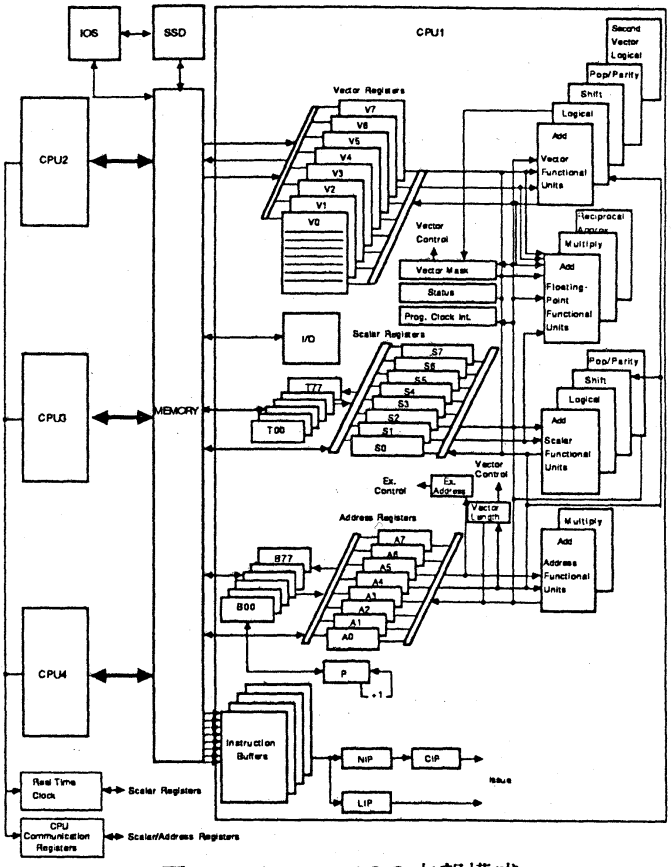
3. CRAY X-MP と CRAY-2

クレイ・リサーチ社では、現在 CRAY X-MPとCRAY-2 の2機種のスーパーコンピュータを製造販売している。 CRAY X-MPは、CRAY-1の後継機種として開発されたものでソフトウェア的には、完全な互換性を保ちながらアーキテクチャの改良により大幅に性能を向上させたものであり、またCRAY-2は、互換性を考慮せずにまったくの新設計、新概念のもとに開発されたものであるがFORTRANレベルでの互換性は保っている。CRAY X-MPとCRAY-2のハードウェア内部構成をそれぞれ図－3と図－4に示す。

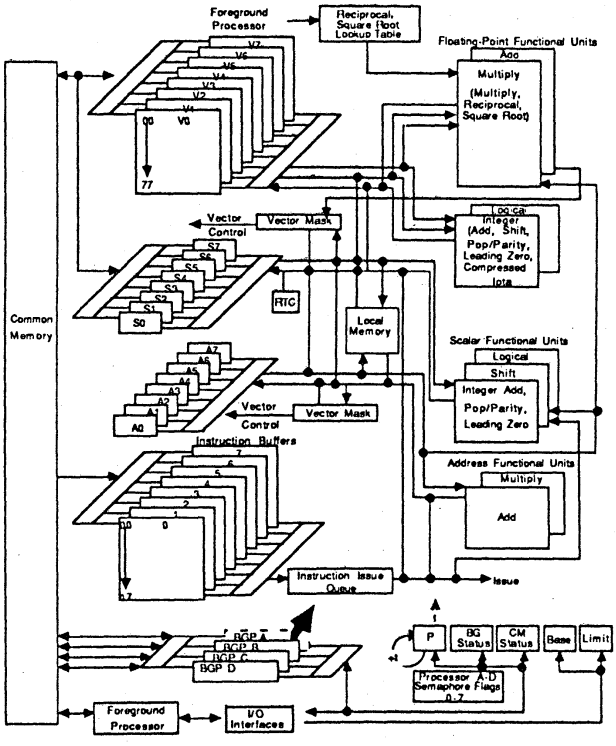
CRAY X-MPとCRAY-2に共通な特徴としては、 1)スカラー演算器の命令機構のみでなく実行部分もパイプライン化した高速スカラー処理、2)短ベクトルにおいて最大性能に近づく立ち上り性能の高いベクトルーパイプライン機構、3)ベクトル性能のみではなくスカラー性能も向上させる主記憶共有型の多重プロセッサ方式 (X-MPが1,2,4CPU構成、CRAY-2が2,4CPU構成)、4)転送速度毎秒12メガバイトの高速磁気ディスク装置の複数台並列動作(ストライピング)による外部入出力があげられる。

CRAY X-MPの特徴としては、上記に加え、1)スカラー,ベクトル演算器ともに8.5ナノ秒のクロック周期、 2)中間レジス

タ(B,Tレジスタ)とパイプライン併用によるスカラー演算,アドレス計算の高速化、3)すべて並列同時動作可能な1CPUあたり14個の演算器、4)アクセス時間34ナノ秒もしくは68ナノ秒の高速主記憶、5)I/Oサブシステム内(IOS)のI/Oプロセッサ(IOP)による入出力処理のCPUからの独立化、6)最大転送速度毎秒2.5ギガバイト、最大記憶容量4ギガバイトの半導体記憶装置(SSD)、7)IOS内の最大転送速度毎秒100メガバイトのバッファメモリ等があげられる。 またCRAY-2の特徴としては、1)液浸冷却方式と高密度実装による4.1ナノ秒のクロック周期、 2)すべて並列同時動作可能な1CPUあたり9個の演算器、




図－4　CRAY X-MPの内部構成

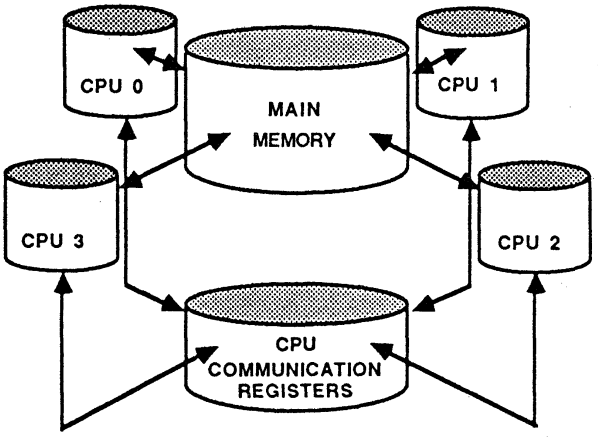



図－5　CRAY-2の内部構成

3)最大256メガ語(2ギガバイト)の実装主記憶、4)1CPUあたり128キロバイトの超高速ローカルメモリ、5)フォアグラウンド・プロセッサによるシステム制御と入出力制御、そして6)全高114センチメートル,直径135センチメートルというコンパクトな筐体があげられる。

主記憶上のプログラムを多重プロセッサにより並列処理するためには、タスクの発生，終了及びタスク間の同期等のためにCPU間の通信が必要となる。CRAY RESEARCH社の多重プロセッサ型スーパーコンピュータでは、いずれのCPUからもアクセス可能な共有コミュニケーションレジスタ(図－5)を装備することにより多重プロセッサの制御及び通信を実現させているがCRAY X-MPとCRAY-2では、その構成も異なる。

X-MPは、CPU数プラス1個の共有レジスタ群(クラスタ)を持っており1クラスタは、CPU間データ転送用の24ビットの共有アドレス・レジスタと64ビットの共有スカラ・レジスタを8個ずつと1ビットのCPU間同期制御用セマフォ・レジスタ32個からなっている。それに対しCRAY-2では、より簡略化されCPU間同期制御のために1ビットのセマフォ・レジスを8個持っているのみである。




図－3　主記憶共有型多重プロセッサの構成

## 4. CRAY RESEARCH社におけるマルチ・タスキング

CRAY RESEARCH社のスーパーコンピュータのマルチ・タスキング手法については、現在のところFORTRANソースプログラムレベルでサブルーチン形式とライブラリ・ルーチンの使用によるマクロ・タスキングとソース内に指示文を挿入することにより前処理される自動形式のマイクロ・タスキングがある。CRAY X-MPでは前者と後者の両方をサポートしており、ひとつのコードにそれらを併用することも可能である。またCRAY-2では現在のところ前者のみをサポートしているが近い将来には、後者もサポートすることと思われる。

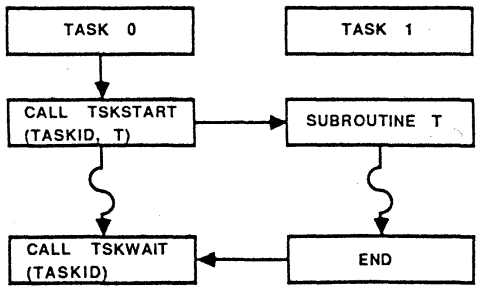
主記憶共有型並列計算機による並列多重処理においてソフトウェア上もっとも留意しなければならないのは、プログラムのメモリー構成である。実行形式のFORTRANプログラムのメモリー領域は、コード領域とデータ領域に分けられ、さらにデータ領域は、各ルーチンに局所的なローカルデータ領域と各ルーチンに共通なコモンデータ領域に分けられる。マルチ・タスキングの実行のためには、ローカルデータ領域は、各々のタスクで独立にする必要があり、タスク発生時に動的に割り当てられる。さらにコモンデータ領域も、タスク間で共有しうるものと独立なものに分けられるが、後者をサポートするためにタスクコモンという概念を導入している。通常のコモンデータ領域がプログラムのローディング時に割り当てられるのに対しタスクコモンデータ領域は、ローカルデー

タ領域と同様にタスク発生時に動的に割り当てられる。

また主記憶共有型並列計算機による並列多重処理を有効に適用するためは、次の事にも注意する必要がある；1)ベクトル化を損なわない，2)並列処理部内のグラニュラリティを大きくする，3)各CPUへの負荷を均等にする，4)並列処理のための前後処理を小さくする。　大規模計算においては、ベクトル長及び並列処理内の計算量の増加により 1),2),4)は、必然的に満たされる場合が多いが、ベクトル処理による加速率が並列処理を上回る現状においては、ベクトル化を優先する必要があり、またそのためにベクトル長の短縮化による並列処理も可能な限り避けることが望ましい。

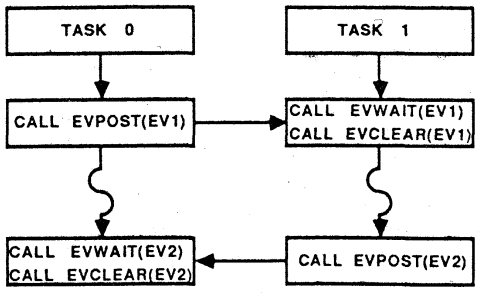
## 4.1 マクロ・タスキング

マクロ・タスキングのためのライブラリには、1)複数のタスクの発生・消滅を制御するためのタスク制御ルーチン、2)タスク間の同期をとるイベント制御ルーチン、3)共通データ領域を他のタスクから保護するロック制御ルーチンの3種のルーチン群がありユーザは、プログラム中でこれらのルーチンを呼ぶことにより、マルチ・タスキングを実現することができる。タスク制御ルーチンには、子タスクを発生さ

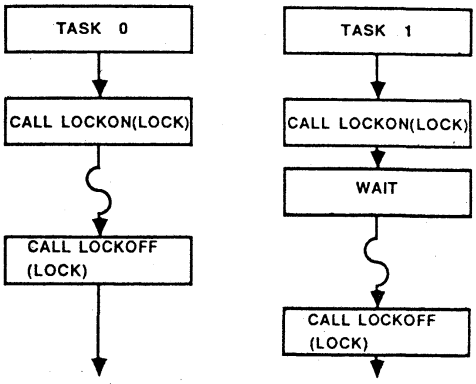



図－6　タスク制御ルーチン

せる"TSKSTART"と子タスクの実行が終了するまで親タスクの実行を待たせる"TSKWAIT"がある(図－6)。イベント制御ルーチンとしては、並行して実行しているタスクに対しイベントを発行する"EVPOST"、イベントが発行されるまで実行を中断する"EVWAIT"、そしてイベントを受けた後それを解除するための"EVCLEAR"がある(図－7)。またロック制御ルーチンには、以後に続く実行文を他のタスクからロックをかける"LOCKON"、及び"LOCKON"でかけたロックを解除する"LOCKOFF"からなる(図－8)。




図－7　イベント制御ルーチン




図－8　ロック制御ルーチン

## 4.2 マイクロ・タスキング

マイクロ・タスキングは、DOループの強制ベクトル化などに用いられるコンパイラ補助制御文(compiler directive)と同様な指示文をFORTRANソースプログラム内に挿入することによりソースプログラムを自動的にマルチ・タスキング用に前処理するもので容易にマルチ・タスキングを使用するすることができるうえに並列処理のためのオーバーヘッドも

非常に低いという特徴をもっている。マイクロ・タスキング用の指示文の使用法を図－9に示す。

```
      PROGRAM  MAIN
      ---------------
CMIC$ GETCPUS(4)          <-- 4個のCPUを設定
      ---------------
      CALL SUB
      ---------------
CMIC$ RELCPUS             <-- CPUを開放
      ---------------
      STOP
      END
C
CMIC$ MULTI               <-- 以下のサブルーチン
      SUBROUTINE  SUB         を並列処理
      ---------------
CMIC$ PROCESS             <-- 並列処理部 -A  開始指定
      (並列処理部 -A:  CPU0)
CMIC$ ALSO PROCESS        <-- 並列処理部 -B  開始指定
      (並列処理部 -B:  CPU1)
CMIC$ END PROCESS         <-- 並列処理部 終了指定
      ---------------
CMIC$ DO  GLOBAL          <-- ループの並列処理指定
      DO  2000  I=1,M
      DO  1000  J=1,N
      ---------------     --> I=1,5,9 ---    : CPU0
      ---------------     --> I=2,6,10 ---   : CPU1
 1000 CONTINUE            --> I=3,7,11 ---   : CPU2
 2000 CONTINUE            --> I=4,8,12 ---   : CPU3
      RETURN
      END
```

図－9　マイクロ・タスキング指示文

## 4.3 数値実験例

CRAY-2にマクロ・タスキングを使用することにより512元の正方行列の積を0.155秒の経過時間で処理することが可能となりグラニュラリティの小さな問題に対してもマルチ・タスキングが有効であることが実証されている。ここでは、マイクロ・タスキングを4CPU構成のCRAY X-MPに適用した例として、1)有限要素法のる平面応力問題の剛性マトリクスの生成、2)改訂コレスキー法，CG法，SCG法による剛性方程式の解法についてスカラー処理、単一プロセッサによるベクトル処理との経過時間の比較を行う。

剛性マトリクスの生成は、1)各要素ごとに要素剛性マトリクスを生成，2)要素剛性マトリクスの全体剛性マトリクスへの加えこみの2段階に分けられる。1)は、要素ごとに独立であり並列処理化が高効率で可能となるのに対して2)は、同一

アドレスへの複数プロセッサによる同時アクセスを発生する可能性があるため処理の順序と同期をとるべき点に注意しなければならない。

図－10に示される改訂コレスキー法のアルゴリズムを検討すると下線部3.0,5.1,7.1がマルチ・タスキングによる並列処理が可能であることがわかる。分解部分3.0は、グラニュラリティが大きくまたベクトル長も短縮する必要がないため高効率の並列処理が可能であるが前進消去部分5.1及び後退代入部分7.1は、内積演算でありベクトル長を短縮化して各CPUに計算を振り分けることにより並列処理が可能となるがグラニュラリティは小さい。しかしながらハイパープレーン法の適用により実際には、簡単なコーディングの変更でループ5.0及び7.0レベルでの並列処理が可能となる。

1.0 $d_{11} = a_{11}$, $l_{11} = 1$
2.0 $l_{i1} = a_{i1}/d_{11}$
3.0 FOR k = 2, ---, n DO
3.1 $u_j^{(k)} = d_{ij}\, l_{kj}$ j=1,---,k-1
3.2 $d_{kk} = a_{kk} - (l_{kj}, u_j^{(k)})$ j=1, —, k-1
3.3 $l_{kk} = 1$
3.4 IF k = n GOTO 4.0
3.5 FOR i = k+1, ---, n DO
3.5.1 $l_{ik} = (a_{ik} - (l_{ij}, u_j^{(k)})/d_{kk}$ j=1, ---, k-1
4.0 $y_1 = b_1$
5.0 FOR k = 2, ---, n DO
5.1 $s_k = (l_{kj}, y_j)$ j=1, ---, k-1
5.2 $y_k = b_k - s_k$
6.0 $x_n = y_n/d_{nn}$
7.0 FOR k = n-1, 1 STEP -1
7.1 $t_k = (l_{jk}, x_j)$ j=k+1, —, n
7.2 $x_k = y_k/d_{kk}$

図－10 改訂コレスキー法

1.0 $r^{(1)} = b - A x^{(1)}$
2.0 $z^{(0)}$ = arbitrary
3.0 FOR k = 1, 2, --- DO
3.1 Solve $M z^{(k)} = r^{(k)}$
3.2 $BS = (z^{(k)}, r^{(k)})$ k>1
BS = 0 k=1
3.3 $b_k = BS/BB$ k>1
3.4 $p^{(k)} = z^{(k-1)} + b_k p^{(k-1)}$
3.5 $q^{(k)} = A p^{(k)}$
3.6 $BB = (p^{(k)}, q^{(k)})$
3.7 $a_k = BS/BB$
3.8 $x^{(k+1)} = x^{(k)} + a_k p^{(k)}$
3.9 $r^{(k+1)} = r^{(k)} - a_k q^{(k)}$
3.10 BB = BS
4.0 END

図－11 前処理付共役勾配法

また共役勾配法(CG法)系のアルゴリズムは一般に図－11に示される下線部1.0,2.0,3.1,3.2,3.4,3.5,3.6,,3.8,3.9と数

多くの部分が並列処理可能であるが各部分のグラニュラリティも小さく3.5が行レベルでの並列性によりベクトル長を変えずに処理できるのを除き、すべてベクトル長の短縮化が必要となる。しかしながら近似解計算部3.8と残差ベクトル計算部3.9は、独立しているため同時に処理することが可能でありベクトル長短縮化の影響を小さくすることができる。CG法系のアルゴリズムは、前処理部分3.1の選択により各種のPCG法が与えられるがここでは、M=Iとした一般的なCG法及びM=Diag(A)としたSCG法のみを用いた。

表－1
ベクトル処理
及び
マルチ・タスキング
の効果

| | N | 512 | 1152 | 2048 | 3200 |
|---|---|---|---|---|---|
| GENERATION | SCALAR | 79.9 (1.0) | 324 (1.0) | 931 (1.0) | 2260 (1.0) |
| | VECTOR | 28 (2.9) | 89 (3.6) | 227 (4.1) | 484 (4.7) |
| | MULTI (4) | 7.3 (11) | 23 (14) | 58 (16) | 123 (18) |
| MODIFIED CHOLESKI | SCALAR | 8674 (1.0) | 95.0K (1.0) | 533K (1.0) | 2006K (1.0) |
| | VECTOR | 711 (12) | 6230 (15) | 25.4K (21) | 87.1K (23) |
| | MULTI (4) | 189 (46) | 1580 (60) | 6390 (83) | 21.8K (92) |
| CONJUGATE GRADIENT | SCALAR | 16.1K (1.0) | 122K (1.0) | 594K (1.0) | 1879K (1.0) |
| | VECTOR | 1749 (9) | 12.1K (10) | 47.9K (12) | 139K (14) |
| | MULTI (4) | 504 (32) | 3370 (36) | 13.1K (45) | 37.7K (50) |
| SCALED CONJUGATE GRADIENT | SCALAR | 11.5K (1.0) | 68.6K (1.0) | 333K (1.0) | 1010K (1.0) |
| | VECTOR | 1246 (9) | 6795 (10) | 26.9K (12) | 76.5K (13) |
| | MULTI (4) | 362 (32) | 1920 (36) | 7480 (45) | 21.0K (48) |

表－1に剛性マトリクスの生成及び各解法の計算時間を経過時間で示す。ここにCG法系の残差2乗和を EPS≦1.0E-03 とした。ベクトル長が一般に短くなりがちの剛性マトリクス生成に対しては、マルチ・タスキングがきわめて有効となることが明らかである。また解法に関しては、大規模行列に対するCG法系の優位性と改訂コレスキー法のベクトル処理及びマルチ・タスク処理時における加速率の高さが目立つ。

## 5.0 おわりに

剛性マトリクスの生成や行列の直接解法のように比較的グラニュラリティが大きく同期点も少ないものだけではなくCG法のようなグラニュラリティが小さく同期点が非常に多い演算に対しても多重プロセッサにより演算時間の大幅な短縮化が可能であることが数値実験により示された。

著者らの経験では構成プロセッサ単体の特性の多重プロセッサに影響をきわめて大きい。コードもしくはアルゴリズムのマルチ・タスク化においては、ベクトルループのベクトル長の短縮化による複数プロセッサへの割り当てが必要となる場合がありベクトル長の短縮によるベクトル演算の効率の低下を最小限にとどめるためには、ベクトル演算器の立ち上り性能が高い事が必須となる。また複数プロセッサによる同じ主記憶バンク等への参照の可能性が単一プロセッサ時に比べて格段に高くなり必然的に起こる待ち時間を最小限にとどめるためには、主記憶のアクセス時間の短縮も必須となる。

現在開発が進められているスーパーコンピュータには、8CPUもしくは16CPU構成で最大性能，実効性能，スカラー性能ともに現在最高水準のものに比べて10倍近いものもあり、それらの主記憶容量も8ギガバイトに及ぶといわれる。そのようなコンピュータを有効に活用するために、よりグラニュラリティが大きく同期点の少ないアルゴリズムの開発及び研究がこれまで以上に重要となってくる。